

ПРАКТИЧНА РОБОТА № 16

Тема: Робота з файлами

Мета: Набути навички роботи з файлами та потоками даних

16.1 Теоретичні відомості

Дивіться лекцію на тему основи роботи з файлами та потоками.

16.2 Виконання завдання № 1

Розглянемо програму з функцією `exponentiation`, яка створює послідовність ступенів числа "2" і виводить їх по одному на рядок у файл. Головна функція передає ім'я файлу у виклику `exponentiation("powers.txt")` і повертає отримане значення (0 або 1). Якщо файлу `powers.txt` у папці з програмою не було, то він створюється. Де Файл заголовків `<iostream>` потрібен для роботи з екраном, `<fstream>` – з потоком, `<math.h>` – для зведення в ступінь.

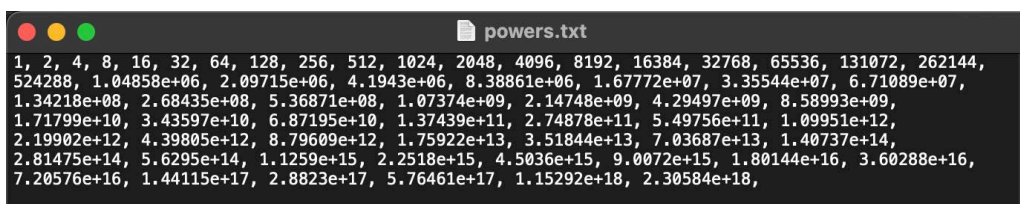
Лістинг 16.1 - Приклад коду програми.

```
#include <iostream>
#include <fstream>
#include <math.h>
using std::cout;
int exponentiation(char *fName) // об'явлення функції
{
    std::ofstream f(fName); // створення екземпляру
    if (!f)
    {
        cout << "Problems with file " << fName;
        return 1;
    }

    int n = 62; // об'явлення змінних
    for (int i = 0; i < n; ++i) //виведення чисел
        f << pow(2, i) << " "; //i кінців рядків !! (1 << i)

    f.close(); // закривається редагування файлу
    return 0;
}
int main() {
    return exponentiation("powers.txt"); //виклик функції
}
```

Результат роботи програми

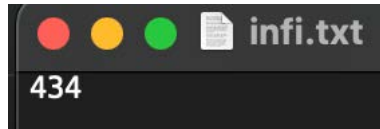
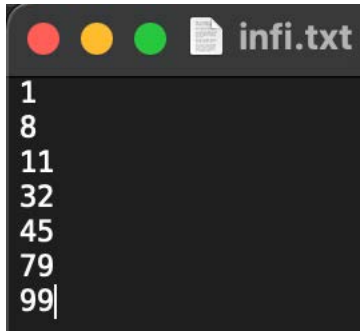


```
1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536, 131072, 262144,
524288, 1.04858e+06, 2.09715e+06, 4.1943e+06, 8.38861e+06, 1.67772e+07, 3.35544e+07, 6.71089e+07,
1.34218e+08, 2.68435e+08, 5.36871e+08, 1.07374e+09, 2.14748e+09, 4.29497e+09, 8.58993e+09,
1.71799e+10, 3.43597e+10, 6.87195e+10, 1.37439e+11, 2.74878e+11, 5.49756e+11, 1.09951e+12,
2.19902e+12, 4.39805e+12, 8.79609e+12, 1.75922e+13, 3.51844e+13, 7.03687e+13, 1.40737e+14,
2.81475e+14, 5.6295e+14, 1.1259e+15, 2.2518e+15, 4.5036e+15, 9.0072e+15, 1.80144e+16, 3.60288e+16,
7.20576e+16, 1.44115e+17, 2.8823e+17, 5.76461e+17, 1.15292e+18, 2.30584e+18,
```

16.3 Виконання завдання №4

Розглянемо головну функцію, яка в циклі вводить числа з файлу infi.txt, "згладжує" та виводить їх до файлу smoothed.txt. Для її компіляції необхідно включити файли заголовків *<iostream>* та *<fstream>*.

Створимо файл з даними «infi.txt».

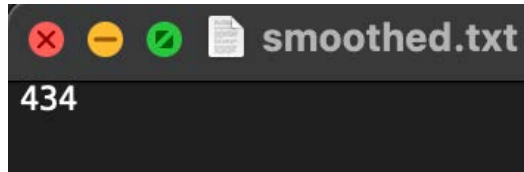
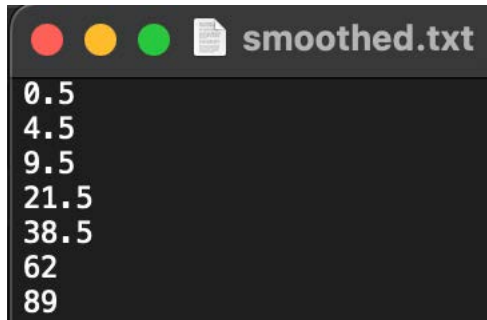


Лістинг 16.2 - Приклад коду програми.

```
#include <iostream>
#include <fstream>
using namespace std;

int main() {
    double cnst, temp = 0, counter = 0; // обявлення змінних
    std::ifstream f("infi.txt"); std::ofstream g("smoothed.txt");
    // перевірка файлів
    if (!f) {
        cout << "Cannot open input file\n";
        return 1;
    } else if (!g) {
        cout << "Problems with output file\n";
        return 1;
    }
    while (f >> cnst) counter++; // підрахунок кількості констант
    if (counter == 0) { // перевірка наявності констант в файлі
        cout << "infi.txt is empty" << endl;
        return 1; }
    f.clear(); f.seekg(0); // повернення до початку зчитування файлу
    //обчислення та заповнення файлу
    while (f >> cnst) {
        if (counter > 1) {
            g << (temp + cnst) / 2 << '\n';
            temp = cnst;
        } else g << cnst << '\n';
    }
    f.close(); g.close(); // закриття читання та редагування файлу
    return 0;
}
```

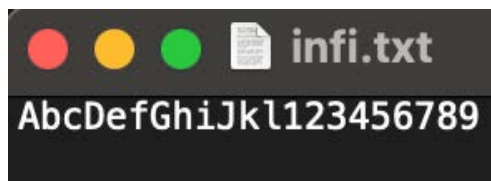
Результати роботи програми



16.4 Виконання завдання №4

Розглянемо головну функцію, яка вводить вміст файлу `infi.txt` та виводить його а) на екран при виклику `screenOutput`; б) в інший файл при виклику `fileOutput`; в) у кінці іншого файлу при виклику `fileEndOutput`. Для її компіляції необхідно включити файли заголовків `<iostream>` та `<fstream>`.

Створимо файл з даними «`infi.txt`».



Лістинг 16.3 - Приклад коду програми.

```
#include <iostream>
#include <fstream>
using namespace std;

int screenOutput (ifstream f) {
    char ch;
    while (f >> ch) cout << ch; /* цикл уведення */
    cout << endl;
    f.close(); // закривається редагування файлу
    return 0;
}

int fileOutput(std::ifstream f, std::ofstream g) {
    char ch;
    while (f >> ch) g << ch; /* цикл уведення */
    f.close(); g.close(); // закривається редагування файлів
    return 0;
}

int fileEndOutput (std::ifstream f, std::ofstream g) {
    char ch;
    g.seekp(0, ios::end); // переміщення курсору на кінець файлу
    g << '\n'; // відступ від попередніх даних
    while (f >> ch) g << ch; /* цикл уведення */
    f.close(); g.close(); // закривається редагування файлів
    return 0;
}
```

```

int main() {
    // перевірка файлів
    if (!std::ifstream("infi.txt")) {
        cout << "Cannot open input file\n"; return 1;
    } else if (!std::ofstream("output.txt")) {
        cout << "Problems with output file\n"; return 1;
    }
    // вибір методу виводу
    cout<<"Choose output method:\n'a'-screen;\n'b'-another file;\n
'c'-end of another file.\n";
    char method;
    cin >> method;
    // виклик функцій
    switch (method) {
        case 'a': return screenOutput(ifstream("infi.txt"));
        case 'b': return fileOutput(std::ifstream("infi.txt"), std::ofstream("output.txt"));
        case 'c':
            return fileEndOutput(std::ifstream("infi.txt"), std::ofstream("output.txt",
ofstream::app));
        default:
            cout << "You have not choosen method." << endl;
            return 1;
    }
}

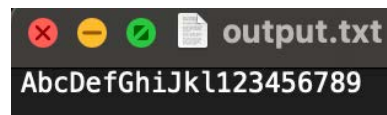
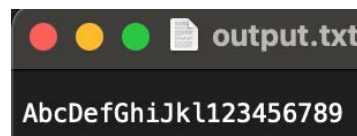
```

Результати роботи програми

```

Choose output method:
'a'-screen;
'b'-another file;
'c'-end of another file.
a
AbcDefGhiJkl123456789
Program ended with exit code: 0

```

16.5 Виконані завдання

Задача 1. Написати програму створення тексту з таблицею степенів числа 2 від 1 до 62.

Задача 4. Текст містить послідовність дійсних констант, що задають числа a_0, a_1, \dots . Про їх кількість відомо, що вона менше максимального значення типу `int`. Увести їх і вивести в інший файл "згладжену" послідовність b_1, b_2, \dots , де $b_1 = (a_0 + a_1)/2$, $b_2 = (a_1 + a_2)/2$, Якщо у вхідній послідовності тільки одне число, то воно й виводиться.

Задача 8. Написати програму виведення вмісту файлу: а) на екран; б) в інший файл; в) у кінці іншого файлу.

16.6 Матеріали та обладнання

Апаратне забезпечення – IBM-сумісний ПК.

Середовище програмування – Xcode.

16.7 Висновки: під час виконання цієї практичної роботи були здобуті навички роботи з файлами та потоками даних.